SOCRATICPO: POLICY OPTIMIZATION VIA INTERACTIVE GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Outcome-based reinforcement learning (RL) improves large language model reasoning by reinforcing correct responses, but leaves failed attempts without a path to correction. Inspired by Socratic teaching, in which a human teacher uses questions and hints to prompt learners to examine and repair their own mistakes, we introduce **SocraticPO** (Socratic Policy Optimization) to model failed reasoning as a teacher–student correction process. After an independent attempt fails, the teacher provides guidance conditioned on that attempt, and the student generates its own revision, shifting learning from outcome selection to guided correction. Rewarding assisted and independent solutions equally can encourage reliance on the teacher. SocraticPO therefore uses assistance-aware, batch-adaptive reward decay to favor solutions requiring less guidance while preserving non-negative normalized advantages for successful corrections. Across four SciKnowEval scientific reasoning domains and GSM8K, SocraticPO outperforms conventional RL and self-distillation baselines in unassisted accuracy under a controllable additional inference budget. Further analyses show that a teacher's answer accuracy does not necessarily predict its guidance quality and that the gains do not require a consistent reduction in teacher–student KL divergence, highlighting the advantage of text-level guidance over answer or distribution transfer. Code is available at `https://anonymous.4open.science/r/SocraticPO-3B7F`.

1 INTRODUCTION

Large language models (LLMs) have advanced rapidly through scaling, instruction tuning, and post-training (Vaswani et al., 2017; Brown et al., 2020; Llama, 2023; Achiam et al., 2023; Bai et al., 2023). Reinforcement learning (RL) has become a central post-training paradigm, optimizing model policies with task rewards or preference-derived signals (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022). In reasoning tasks, these rewards are often outcome based and verifiable, such as binary answer correctness.

However, outcome-based RL learns primarily through *selection*. Successful rollouts are reinforced, while unsuccessful ones are suppressed. When a response is wrong, its reward indicates that the sampled behavior should become less likely, but does not turn the failed attempt into an example of how to reason correctly (Cai et al., 2025).

Human teaching treats mistakes differently. Instead of providing only a grade, a teacher inspects the learner's attempt and offers a question, hint, or critique that helps the learner revise their reasoning (Wood et al., 1976; VanLehn, 2011). In the Socratic tradition, such questions invite the learner to identify and repair a misconception rather than copy a supplied answer (Scott, 2006; Liu et al., 2024). Guided revision is a productive way for people to learn from mistakes (VanLehn, 2011). This raises a question for LLM training: *can we use a teacher model to help a student policy learn from how its own failed reasoning is corrected?*

Inspired by this principle, we introduce **Socratic Policy Optimization (SocraticPO)**, which models the teacher–student correction process within policy optimization. The student first attempts a problem independently. If its answer is wrong, the teacher examines that attempt and points to a specific contradiction, misconception, or next step through a concise hint. The student must reconsider its original reasoning and generate the revised response itself. Figure 1 illustrates this sequence

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

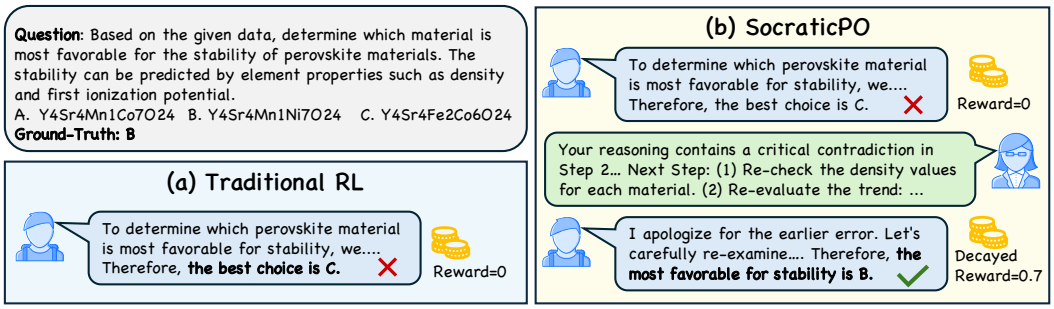

Figure 1: From outcome selection to guided correction on a materials-science question. (a) Conventional RL assigns zero reward to an incorrect student response and ends the rollout. (b) SocraticPO uses that failed attempt to elicit a diagnostic hint; the student generates its own correction, and the assisted success receives a decayed reward.

on a materials-science question. The first answer selects the wrong material; guidance asks the student to revisit the property values and trend behind that choice. Notably, SocraticPO differs from distillation in requiring only text-level guidance rather than teacher-generated target responses or probability distributions (Hinton et al., 2015; Agarwal et al., 2024; Fu et al., 2026). It can therefore use black-box teachers without access to their logits.

Introducing guidance creates a new optimization challenge. If teacher-assisted and independent solutions receive equal rewards, the policy may learn to rely on intervention rather than internalize the correction. SocraticPO therefore pairs guidance with **batch-adaptive, assistance-aware reward decay**. We prove that in this setting, solutions reached with less guidance can receive greater credit while successful corrections retain non-negative normalized advantages. This lets failed attempts contribute to learning while discouraging dependence on teacher help.

We evaluate SocraticPO on four undergraduate-level scientific reasoning domains from SciKnowEval (Feng et al., 2024) and on GSM8K math problems (Cobbe et al., 2021). Across these benchmarks, SocraticPO improves unassisted accuracy over conventional RL baselines under a controllable additional inference budget. Further analyses reveal that answer accuracy does not necessarily predict guidance quality, that giving the teacher ground-truth answers even can weaken learning through answer leakage.

Our main contributions are:

- **Learning through correction.** We model teacher–student interaction within policy optimization, turning failed rollouts into guided correction trajectories in which the student produces its own repair while remaining compatible with existing policy-gradient backends.
- **Assistance-aware reward design.** We introduce reward decay to successful corrections while discouraging dependence on assistance, and characterize its normalized-advantage properties.
- **Black-box teacher compatibility.** SocraticPO requires only text-level guidance rather than distribution matching, allowing stronger API-based or otherwise black-box teachers to participate in policy optimization.
- **Empirical evidence and analysis.** We show higher unassisted accuracy on scientific reasoning and math benchmarks, isolate the roles of targeted feedback and reward decay, and identify distinctions between teacher knowledge, guidance quality, and distribution matching.

2 Related Work

2.1 Outcome and Process Supervision for Reasoning

Reinforcement learning has been used to improve LLMs using human preferences, AI feedback, and verifiable task rewards (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022). In reasoning tasks, policy-gradient methods such as PPO, REINFORCE++, and GRPO use these rewards to favor responses with higher estimated advantage (Schulman et al., 2017; Williams, 1992; Sutton et al., 1999; Hu et al., 2025; Shao et al., 2024).

However, an outcome reward only indicates whether a response is correct, not where its reasoning went wrong or how the student could revise it. Process supervision provides feedback on intermediate steps and can help identify errors in the reasoning (Uesato et al., 2022). Yet evaluating an existing trajectory does not give the student a chance to revise that trajectory within the same rollout. SocraticPO turns diagnosis into an opportunity for correction: the teacher provides guidance conditioned on the failed attempt, the student revises its reasoning, and the task verifier evaluates the new response. When the revision succeeds, that response becomes a positive training signal.

2.2 Interactive Feedback and Socratic Teaching

Educational research distinguishes evaluation from instruction. Tutors scaffold learning by inspecting a student's current attempt, diagnosing misconceptions, and providing hints or next-step guidance (Wood et al., 1976; VanLehn, 2011). The Socratic tradition similarly uses questions and critiques to elicit revision instead of directly transferring an answer (Scott, 2006). Liu et al. (2024) explore LLM-based Socratic tutoring and personalized teaching dialogue.

SocraticPO brings this teaching pattern into policy optimization. The teacher intervenes only after the student gives an incorrect answer, tailoring a hint to the mistake in that answer. The student then produces a revised response, whose correctness is determined by the task verifier. This differs from teacher-generated demonstrations and distribution-matching distillation, which treat teacher behavior as an imitation target (Hsieh et al., 2023; Agarwal et al., 2024; Hübotter et al., 2026).

3 Preliminaries

We first describe the outcome-based RL setting underlying the distinction between selecting responses and learning to correct them. Given a prompt x, an autoregressive policy π_θ generates a response $y = (y_1, \ldots, y_T)$ with probability

$$\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}). \tag{1}$$

Let $\mathcal{G}$ denote verifier-side information, such as a reference answer, and let $\delta(y, \mathcal{G}) = \mathbb{I}[\text{correct}(y, \mathcal{G})]$ indicate correctness. In single-turn outcome-based RL, a response is a complete rollout and receives a scalar reward $R(y)$; for binary correctness, $R(y) = \delta(y, \mathcal{G})$.

For a fixed prompt, the standard objective is expected reward (Williams, 1992; Sutton et al., 1999).

$$\mathcal{J}(\theta) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[R(y)]. \tag{2}$$

Policy-gradient methods weight the score $\nabla_\theta \log \pi_\theta(y \mid x)$ by an estimated advantage, often using baselines, normalization, or clipping to stabilize training (Schulman et al., 2017; Shao et al., 2024; Hu et al., 2025).

An incorrect response can therefore affect the update, for example through a negative advantage. What this single-turn setting does not provide is a continuation that shows the student how to revise that particular attempt. The opportunity is to make the failure a starting point for further reasoning, rather than only an outcome to score.

4 Method

SocraticPO treats a failed attempt as the beginning of a learning interaction. The teacher responds to the student's actual reasoning with a diagnostic question or next-step hint, so that the student must produce the repair rather than copy an answer. Learning from that repair should also favor eventual independence. Interactive rollouts supply the targeted guidance, and assistance-aware rewards favor corrections made with less help.

4.1 Interactive Rollout

A failed answer reveals the line of reasoning the student actually pursued. To make that information available for diagnosis, the student first attempts the problem without help. Guidance is introduced

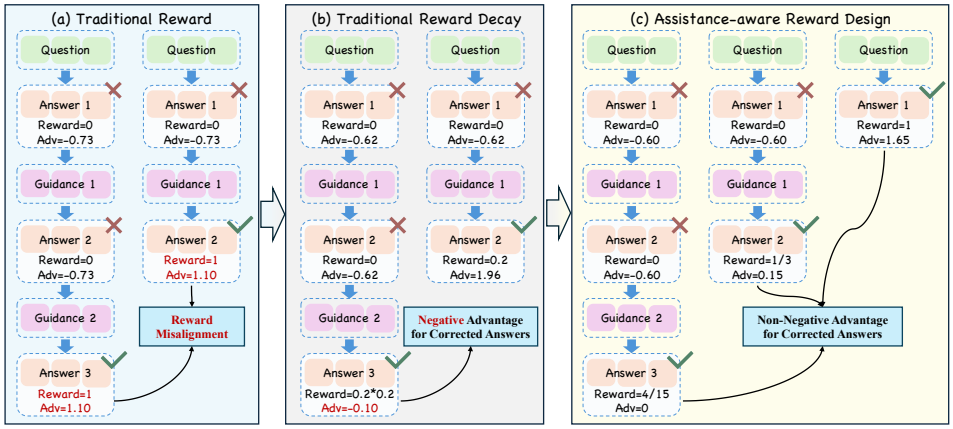

Figure 2: Assistance-aware credit assignment across correction trajectories. (a) Binary rewards give equal credit to successful answers regardless of how much guidance they require. (b) Fixed exponential decay can push a later correct answer below the batch mean, yielding negative normalized advantage. (c) SocraticPO sets assisted rewards from the preceding smoothed mean, preserving non-negative advantage for corrected answers while favoring earlier success.

only if the attempt fails. We allow at most K complete student attempts and start from $c^{(1)} = x$. The index k counts attempts, not individual reasoning steps or tokens. On attempt k, the student generates a response and the verifier evaluates it:

$$y^{(k)} \sim \pi_\theta(\cdot \mid c^{(k)}), \qquad \delta^{(k)} = \delta(y^{(k)}, \mathcal{G}). \tag{3}$$

A correct response terminates the rollout. After a failure, the outcome score alone leaves the student without a direction for revision. If $k < K$, we therefore query a teacher π_ϕ for guidance conditioned on the attempt and any teacher-side information $\mathcal{I}$:

$$\begin{aligned} f^{(k)} &\sim \pi_\phi(\cdot \mid c^{(k)}, y^{(k)}, \mathcal{I}), \\ c^{(k+1)} &= c^{(k)} \| y^{(k)} \| f^{(k)}, \end{aligned} \tag{4}$$

Here $\|$ denotes concatenation. The teacher is prompted to identify a flaw in the attempted reasoning or suggest what to reconsider next, leaving the student to produce the revised answer. Teacher-side information is specified separately from verifier-side information.

The resulting trajectory is $\tau = (x, y^{(1)}, f^{(1)}, \ldots, f^{(L-1)}, y^{(L)})$, where $L \leq K$ is determined by the first success or the attempt limit. Retaining both the failed response and its feedback makes the next attempt a targeted revision rather than a blind retry. When that revision succeeds, the student has generated a rewardable response grounded in its own earlier mistake. Appendix B.2 gives the teaching prompt, and Appendix B.1 shows an example for interactive guidance.

4.2 Assistance-Aware Rewards

Guided correction should help the student solve with less assistance, not simply make assisted answers easier to reward. Figure 2(a) illustrates the problem with binary rewards: solutions obtained after one and two rounds of guidance both receive reward 1. The reward does not distinguish how much help was needed(Wang et al., 2026). Here, we score individual attempts and discount success reached after more guidance, so that earlier solutions receive greater credit.

A natural choice is fixed exponential decay, $r^{(k)} = \gamma^{k-1}\delta^{(k)}$ with $0 < \gamma < 1$. However, decreasing the reward is not sufficient: the policy update depends on the reward relative to the batch mean, not on its absolute value. In Figure 2(b), $\gamma = 0.2$ gives the two corrected responses rewards 0.2 and 0.04. Across all five student attempts, the mean is $(0.2 + 0.04)/5 = 0.048$. The later correct response therefore has a negative normalized advantage of approximately -0.10, despite solving the problem. Thus, fixed decay can turn a successful repair into a response the update suppresses. Merely choosing a decay factor does not guarantee that corrected answers retain non-negative credit.

Our design must satisfy both requirements: favor solutions needing less help, while preserving non-negative credit for successful corrections. Since the latter depends on the batch baseline, we set each assisted success's reward using a smoothed mean of the preceding attempts. Let $\mathcal{A}_k$ contain the samples still unsolved before step k, and write $\delta_i^{(k)} = \delta(y_i^{(k)}, \mathcal{G}_i)$. With a smoothing pseudocount $\beta > 0$, set $\mu_0 = 1$ and, for $k \geq 2$, compute

$$\mu_{k-1} = \frac{\beta + \sum_{j=1}^{k-1} \sum_{i \in \mathcal{A}_j} r_i^{(j)}}{\beta + \sum_{j=1}^{k-1} |\mathcal{A}_j|}. \tag{5}$$

We assign

$$r_i^{(k)} = \mu_{k-1} \delta_i^{(k)}, \qquad i \in \mathcal{A}_k. \tag{6}$$

Equation equation 6 ties the reward for a successful correction to the preceding smoothed batch mean. Because every new reward is either zero or that mean, the cumulative mean cannot increase. Successful corrections therefore cannot fall below the later baseline. Failures lower the reward scale, whereas a fully successful round leaves it unchanged. These consequences of the reward rule yield the following properties of normalized advantages.

Theorem 1 (Assistance-Aware Advantage under Reward Decay)**.** *Let μ_t and σ_t be the weighted mean and standard deviation of the rewards collected through step t, augmented by a reward-1 pseudo-observation with weight β. Thus, μ_t follows Equation equation 5. For a response at step $k \leq t$, define*

$$\hat{a}_i^{(k)} = \frac{r_i^{(k)} - \mu_t}{\sigma_t + \epsilon}, \qquad \epsilon > 0. \tag{7}$$

With $\beta > 0$ and $\mu_0 = 1$, the following properties hold:

1. ***Non-negativity.*** *A correct response at step $k \leq t$ has $\hat{a}_i^{(k)} \geq 0$.*
2. ***Monotonicity.*** *For correct responses at steps $k < l \leq t$, $\hat{a}_i^{(k)} > \hat{a}_j^{(l)}$.*
3. ***Saturation decay.*** *Let $p_t = |\mathcal{A}_t|^{-1} \sum_{i \in \mathcal{A}_t} \delta_i^{(t)}$. For a correct response at step t, $\hat{a}_i^{(t)} \to 0$ as $p_t \to 1$.*

The proof is in Appendix A. These properties preserve non-negative credit for successful corrections while favoring less assistance under the stated conditions.

Figure 2(c) provides a concrete illustration in the unsmoothed limit $\beta \to 0$. The three questions are solved at steps 1, 2, and 3, receiving rewards 1, $1/3$, and $4/15$; the final mean over all six attempts is $4/15$. Thus all correct responses have non-negative advantages, with earlier successes receiving larger ones. The final response has zero advantage because the last round corrects its only active sample, corresponding to the $p_3 = 1$ case of saturation decay. Whether these credit-assignment properties translate into stronger independent reasoning is evaluated in Section 5.3.

4.3 Policy Update

With guided repairs available alongside independent successes, both can contribute to policy learning according to the assistance they required. After rollout, rewards are normalized using Equation equation 7, with t the last sampled round. Let $\mathcal{T}$ contain the collected attempts and $M = |\mathcal{T}|$. The attempt-level policy-gradient direction is

$$\widehat{g} = \frac{1}{M} \sum_{k=1}^{t} \sum_{i \in \mathcal{A}_k} \hat{a}_i^{(k)} \nabla_\theta \log \pi_\theta(y_i^{(k)} \mid c_i^{(k)}). \tag{8}$$

Contexts and advantages are held fixed because credit belongs to each response's outcome rather than the full dialogue's return-to-go. We use Reinforce++ as the optimization backend (Hu et al., 2025). Since the teacher's role is to elicit a revision, not to provide an imitation target, its feedback enters only as context and its tokens are excluded from the update. Black-box teacher compatibility follows directly because no teacher likelihoods or distribution-matching loss are needed. Algorithm 1 summarizes the procedure.

Algorithm 1: SocraticPO with Guided Correction and Reward Decay

Input: Student π_θ; teacher π_ϕ; dataset $\mathcal{D}$ with verifier information $\mathcal{G}$ and teacher information $\mathcal{I}$; maximum attempts K; batch size B; smoothing pseudocount $\beta > 0$.
Output: Optimized student π_θ.
repeat Sample $\{(x_i, \mathcal{G}_i, \mathcal{I}_i)\}_{i=1}^{B} \sim \mathcal{D}$;
Initialize $c_i^{(1)} \leftarrow x_i$, $\mathcal{A}_1 \leftarrow \{1, \ldots, B\}$, and $\mathcal{T} \leftarrow \emptyset$;
for $k = 1$ **to** K **do**
  foreach $i \in \mathcal{A}_k$ **do**
    Generate $y_i^{(k)} \sim \pi_\theta(\cdot \mid c_i^{(k)})$ and evaluate $\delta_i^{(k)} \leftarrow \delta(y_i^{(k)}, \mathcal{G}_i)$;
    Store $(c_i^{(k)}, y_i^{(k)}, \delta_i^{(k)})$ in $\mathcal{T}$;
    if $\delta_i^{(k)} = 0$ ***and*** $k < K$ **then**
      Generate $f_i^{(k)} \sim \pi_\phi(\cdot \mid c_i^{(k)}, y_i^{(k)}, \mathcal{I}_i)$;
      Set $c_i^{(k+1)} \leftarrow c_i^{(k)} \| y_i^{(k)} \| f_i^{(k)}$;
  Compute $\{r_i^{(k)} : i \in \mathcal{A}_k\}$ using Equation equation 6;
  Set $\mathcal{A}_{k+1} \leftarrow \{i \in \mathcal{A}_k : \delta_i^{(k)} = 0\}$;
  if $\mathcal{A}_{k+1} = \emptyset$ **then**
    break;
Update π_θ with Reinforce++ using student attempts in $\mathcal{T}$ and their rewards;
until convergence

5 Experiments

We evaluate whether guided correction of failed rollouts improves the student's unassisted reasoning and how the teacher and the number of correction attempts affect this gain. We then examine the contributions of targeted guidance and assistance-aware rewards, and analyze the role of teacher information and imitation. Finally, we discuss the training-time cost of additional interaction rounds.

5.1 Experimental Setup

Models. We use **Qwen3-4B-Instruct** as the student policy π_θ. Guidance comes from the student itself, **Qwen3.5-27B**, or **Gemma3-27B**. Self-guidance tests whether the interaction adds value without a larger teacher; the two external teachers let us examine how guidance quality varies across models. Unless otherwise noted, we allow 2 student attempts and 1 teacher guidance ($K = 2$). Full implementation details and prompts are provided in Appendix B.

Datasets and metric. We evaluate on five reasoning benchmarks. Following the SDPO evaluation setup, **Chemistry**, **Physics**, **Biology**, and **Material** come from the undergraduate-level reasoning subsets of **SciKnowEval** (Feng et al., 2024). For **Math**, we use **GSM8K** (Cobbe et al., 2021). For each test example, we sample 16 unassisted responses and average their correctness (mean@16).

Baselines. We compare against **Reinforce++**, **GRPO**, **GSPO**, and **SDPO** (Hu et al., 2025; Shao et al., 2024; Chujie, 2025; Hübotter et al., 2026). Reinforce++, GRPO, and GSPO use scalar-reward RL, while SDPO is a self-distillation baseline. We also report the untrained student and the standalone teacher models as references; these rows are not student-training baselines.

5.2 Main Experimental Results

Table 1 compares unassisted accuracy across benchmarks, while Figure 3 shows how guided and unassisted accuracy evolve during training. Together, they reveal the following patterns.

Guided correction improves unassisted accuracy. All SocraticPO variants exceed the best baseline average, Reinforce++ at 0.7807. SocraticPO reaches 0.8113 with Gemma3-27B guidance, a gain of 3.06 percentage points. Even self-guidance reaches 0.7895 at $K = 2$ and 0.8066 at $K = 3$. The additional round improves the self-guided five-benchmark average by 1.71 points, with clear

Table 1: Unassisted accuracy on four SciKnowEval domains and GSM8K (Math). SocraticPO variants specify the teacher model and maximum number of student attempts K in parentheses. Bold and underline mark the best and second-best result in each column.

Method	Chemistry	Physics	Biology	Material	Math	Avg.
Gemma3-27B	0.4625	0.5866	0.3567	0.6987	0.9469	0.6103
Qwen3.5-27B	0.6931	0.7617	0.2947	0.7453	0.9492	0.6888
Qwen3-4B-Instruct	0.5071	0.6664	0.2762	0.7413	0.8893	0.6161
+Reinforce++	0.7270	0.7726	0.6575	0.7978	0.9484	0.7807
+GRPO	0.7657	0.7773	0.6137	0.7712	0.9500	0.7756
+GSPO	0.7633	0.7697	0.5875	0.7849	0.9492	0.7709
+SDPO	<u>0.7961</u>	0.7742	0.5987	0.7779	0.8893	0.7672
+SocraticPO (Qwen3-4B-Instruct,K=2)	0.7672	0.7648	<u>0.6612</u>	0.8058	0.9484	0.7895
+SocraticPO (Qwen3-4B-Instruct,K=3)	0.7704	<u>0.8281</u>	0.6338	**0.8470**	**0.9537**	<u>0.8066</u>
+SocraticPO (Qwen3.5-27B,K=2)	0.7782	0.7789	**0.6787**	0.8138	0.9438	0.7987
+SocraticPO (Gemma3-27B,K=2)	**0.8023**	**0.8521**	0.6288	<u>0.8217</u>	<u>0.9514</u>	**0.8113**

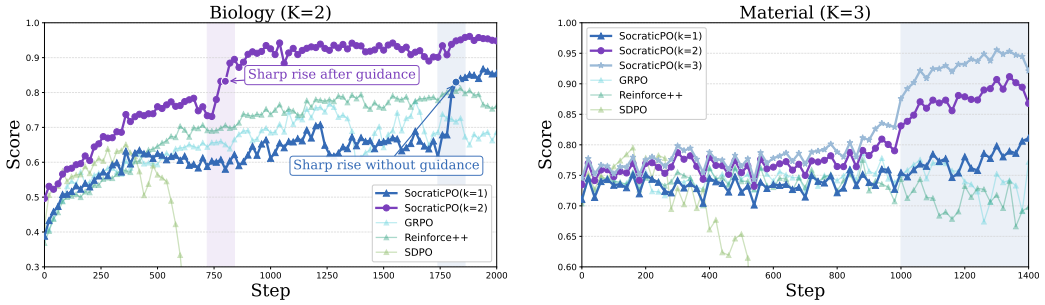

Figure 3: Training accuracy on Biology (left, $K = 2$) and Material (right, $K = 3$), with Qwen3-4B-Instruct as both student and teacher. SocraticPO curves show accuracy on the first unassisted attempt ($k = 1$) and after one or two rounds of guidance ($k = 2, 3$).

gains on Physics and Material but a decline on Biology. Figure 3 suggests why the extra round helps unevenly: on Biology, accuracy after one guidance round is already near ceiling, leaving few failures for a second round; on Material, one-round guidance yields only modest gains early in training, leaving more room for further correction. This suggests choosing the guidance budget according to how many failures remain after the first round, rather than increasing K uniformly across tasks.

External teachers usually improve guidance. Replacing self-guidance with Qwen3.5-27B or Gemma3-27B raises average accuracy from 0.7895 to 0.7987 or 0.8113, respectively. The gains are not uniform: Qwen3.5-27B gives the highest Biology score, whereas Gemma3-27B is stronger on Chemistry and Physics. Thus, guidance from a larger external teacher can improve correction beyond what the student provides for itself, but no teacher dominates every benchmark.

Knowledge and guidance ability do not coincide. Across all five benchmarks, the model with higher standalone accuracy among Qwen3.5-27B and Gemma3-27B is the less effective teacher: its guidance yields lower unassisted student accuracy after SocraticPO training. This reversal motivates dedicated benchmarks that assess teachers by how much their feedback improves student learning, rather than by their own answer accuracy.

Correction precedes independent improvement. Figure 3 traces training on Biology and Material. These curves are suggestive of a staged learning pattern: accuracy with guidance improves before unassisted accuracy, which is consistent with the student first learning to use a correction and later solving more problems without one.

5.3 Ablation Studies

We test separately whether the gain comes from *targeted feedback after a failed attempt* and whether *assistance-aware credit* turns that feedback into unassisted performance. The **w/o tg** variant replaces attempt-conditioned teacher guidance with a fixed retry prompt. It keeps the extra student attempt but removes information about the specific error. The **w/o rd** variant keeps guidance yet gives full

Table 2: Unassisted accuracy in the guidance and reward ablations. **w/o tg** replaces attempt-conditioned teacher guidance with a fixed retry prompt; **w/o rd** removes reward decay; **0.8 rd** uses a fixed decay factor of 0.8.

Teacher	Method	Chemistry	Physics	Biology	Material	Math	Avg.
–	w/o tg	0.7110	0.7703	0.5737	0.7706	0.9487	0.7549
Qwen3-4B-Instruct	w/o rd	0.7586	0.7468	0.5837	0.7859	0.9454	0.7641
	0.8 rd	0.7661	0.7753	0.6400	0.7812	**0.9492**	0.7824
	–	0.7672	0.7648	0.6612	0.8058	0.9484	0.7895
Qwen3.5-27B	w/o rd	0.7053	0.7484	0.5700	0.7992	0.9423	0.7530
	0.8 rd	0.7487	0.7695	0.5909	0.7945	0.9454	0.7698
	–	**0.7782**	**0.7789**	**0.6787**	**0.8138**	0.9438	**0.7987**

reward to every correct answer, regardless of how much help was needed. Finally, **0.8 rd** replaces our batch-adaptive reward with a fixed exponential discount of 0.8. Together, these variants separate the value of diagnosis from the way successful revisions are credited.

Targeted guidance versus blind retrying. The fixed retry prompt reaches 0.7549 average accuracy across the four science domains, below the full SocraticPO variants on those same domains (0.7895 with self-guidance and 0.7987 with Qwen3.5-27B). Merely giving the policy another attempt is therefore insufficient: the intervening text must carry information about the error that occurred.

Assisted success versus independent success. Removing reward decay lowers the five-benchmark average from 0.7895 to 0.7641 with self-guidance and from 0.7987 to 0.7530 with Qwen3.5-27B. As Figure 4 illustrates, full reward can rapidly improve performance in teacher-assisted contexts without producing comparable unassisted gains. A fixed 0.8 decay partially recovers the five-benchmark average to 0.7824 and 0.7698, but remains below the batch-adaptive rule. The sharper degradation with Qwen3.5-27B is especially revealing: guidance increases the opportunity for correction, but also increases the need to discount solutions made easy by that guidance. These results support reward decay as the mechanism that converts useful assistance into pressure for earlier, independent success.

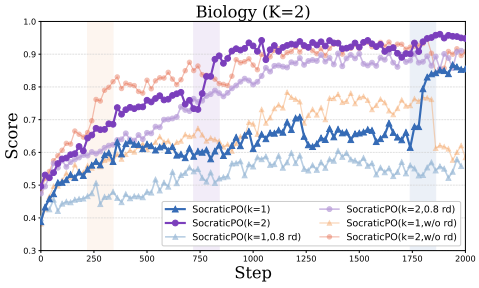

Figure 4: Effect of reward decay on Biology with self-guidance ($K = 2$).

5.4 Further Discoveries

Teacher access to ground truth. We next discuss what information helps a teacher diagnose failure. Although access to the ground-truth answer can make feedback more precise, Figure 5 shows that it degrades student performance in most of our settings. One explanation is answer leakage: even when instructed not to reveal the solution, the teacher may provide hints that make the next response too easy. Ground truth can also shift the task from analyzing the student's reasoning to reverse-engineering a path from the known answer. Both behaviors weaken the distinction between guided repair and answer transfer. Ground-truth information is not intrinsically harmful. The Qwen3-4B-Instruct result on Chemistry suggests that it can help when the teacher uses the reference to diagnose an error without revealing the solution. The relevant variable is therefore not access alone,

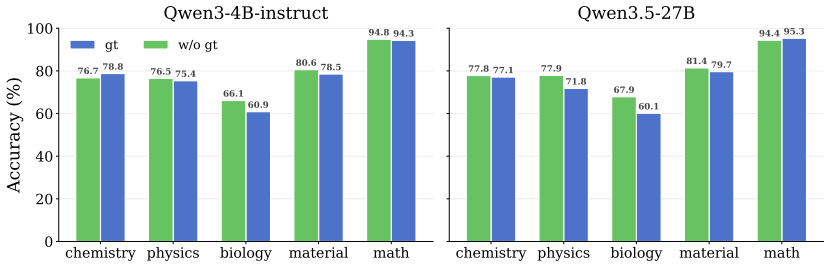

Figure 5: Unassisted accuracy with (gt) and without (w/o gt) teacher access to ground truth. The left panel uses Qwen3-4B-Instruct self-guidance; the right uses Qwen3.5-27B guidance.

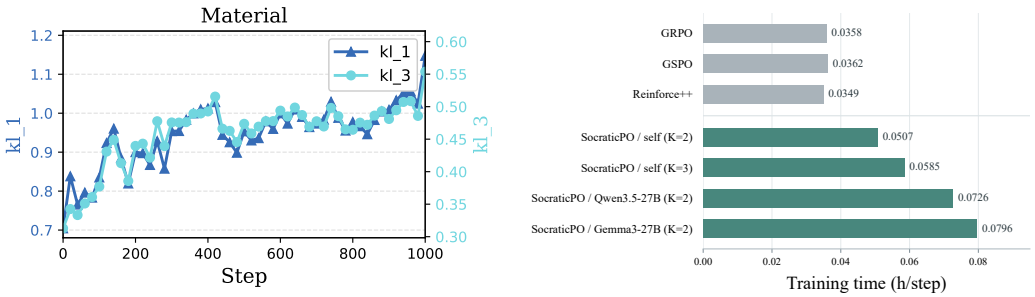

Figure 6: Teacher–student divergence and training-time cost. Left: KL_1 and KL_3 between the Qwen3-8B and Qwen3-4B-Instruct on Material over training steps. Right: wall-clock training time (h/step) for RL baselines and SocraticPO with different teachers and maximum attempt counts K.

but whether privileged information improves diagnosis or bypasses the student's repair process. **Guidance versus imitation.** To test whether SocraticPO is implicit distillation, we measure teacher–student KL divergence during training. We use **Qwen3-8B** as the teacher and **Qwen3-4B-Instruct** as the student on Material because their compatible tokenizers permit token-level comparison. If improvement came primarily from imitation, teacher–student divergence should decrease as task performance rises. Figure 6 (left) shows no consistent KL reduction. This result does not rule out all behavioral transfer, but it supports the intended distinction: teacher shapes the context in which the student repairs an error, while the objective never asks the student to match the teacher's distribution.

Training-time cost. Figure 6 (right) compares training time per step across methods. Reinforce++ takes 0.0349 h/step, compared with 0.0507 h/step for self-guided SocraticPO at $K = 2$ and 0.0585 h/step at $K = 3$. This increase includes additional student rollouts on failed questions, which is intrinsic to the method. Teacher guidance adds a separate deployment-dependent cost. With Qwen3.5-27B and Gemma3-27B teachers, we observe 0.0726 and 0.0796 h/step at $K = 2$, respectively; teacher-side latency depends on model size and allocated serving resources. When teacher inference is a bottleneck, additional teacher replicas can process feedback requests in parallel and substantially reduce wall-clock waiting time, though they cannot remove the student-rollout cost or the total teacher compute. Thus, K controls the necessary revision work, while teacher provisioning controls much of the remaining latency.

6 Conclusion

Outcome-based RL can reinforce successful responses, but a failed rollout usually ends without showing the policy how its particular reasoning attempt could be repaired. We introduced **SocraticPO** to turn this missing transition into a training signal. The student first attempts a problem independently; after a failure, a teacher examines that attempt and provides a diagnostic question or hint, leaving the student to generate its own revision. This makes a successful correction a learnable student response rather than a teacher answer to imitate. Yet rewarding assisted and independent successes equally would encourage reliance on help. SocraticPO therefore uses batch-adaptive, assistance-aware reward decay to give more credit to solutions reached with fewer interventions while retaining non-negative normalized advantages for successful corrections. Because guidance enters as text-level context rather than a target distribution, the method remains compatible with black-box teachers and standard policy-gradient backends.

Across four SciKnowEval science domains and GSM8K, every SocraticPO variant exceeds the strongest baseline in average unassisted accuracy; with Gemma3-27B guidance, the average reaches 0.8113, 3.06 percentage points above Reinforce++. Ablations further show that targeted guidance and assistance-aware reward decay are both important: blind retrying, removing decay, and using a fixed discount all underperform the full method. These results support learning from guided correction while preserving the incentive to solve problems independently.

Although SocraticPO effectively turns guided correction into improved unassisted performance, it still depends on the teacher's ability to follow instructions, while teacher guidance increases training time. Future work will therefore explore more reliable and efficient teacher-guidance architectures.

AI USE STATEMENT

In this work, we used generative AI tools only to polish the language and improve readability. We reviewed all AI-assisted work. In particular, all AI-assisted language edits were reviewed by the authors to ensure that they did not alter the technical meaning or scientific claims of the manuscript. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects or private data. Our experiments use publicly available scientific reasoning and mathematics benchmarks together with pretrained language models. Like other methods that learn from model-generated feedback, SocraticPO may inherit factual errors or biases from its teacher model. Teacher guidance and model outputs should therefore be independently verified before use in high-stakes scientific or educational settings.

REPRODUCIBILITY STATEMENT

We provide the complete training procedure, objective functions, theoretical analysis, and evaluation protocol in the main paper. The appendix further reports the prompts, dataset statistics and examples, implementation details, hardware configuration, and hyperparameters for all compared methods. Evaluation accuracy is computed from 16 independently sampled responses per problem. An anonymized implementation is available at the URL given in the abstract.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Xin-Qiang Cai, Wei Wang, Feng Liu, Tongliang Liu, Gang Niu, and Masashi Sugiyama. Reinforcement learning with verifiable yet noisy rewards under imperfect verifiers. *arXiv preprint arXiv:2510.00915*, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Zheng Chujie. Group sequence policy optimization. *arXiv preprint*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the association for computational linguistics: ACL 2023*, pp. 8003–8017, 2023.

Jian Hu et al. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 1(4), 2025.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. Socraticlm: Exploring socratic personalized teaching with large language models. *Advances in Neural Information Processing Systems*, 37:85693–85721, 2024.

Hugo Touvron Llama. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2407.21783*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Dominic Scott. *Plato's Meno*. Cambridge University Press, 2006.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Kurt VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational psychologist*, 46(4):197–221, 2011.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Daoyu Wang, Qingchuan Li, Mingyue Cheng, Jie Ouyang, Shuo Yu, Qi Liu, and Enhong Chen. Steppo: Step-aligned policy optimization for agentic reinforcement learning. *arXiv preprint arXiv:2604.18401*, 2026.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

David Wood, Jerome S Bruner, and Gail Ross. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17(2):89–100, 1976.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Proof of Theorem 1

Proof. We prove the three properties in Theorem 1. Let $m_k = |\mathcal{A}_k|$ and

$$p_k = \frac{1}{m_k} \sum_{i \in \mathcal{A}_k} \delta_i^{(k)}. \tag{9}$$

Define $N_k = \sum_{h=1}^{k} m_h$, with $N_0 = 0$. With smoothing pseudocount $\beta > 0$ and $\mu_0 = 1$, the smoothed mean reward after including the samples active at step k is

$$\mu_k = \frac{(\beta + N_{k-1})\mu_{k-1} + \sum_{i \in \mathcal{A}_k} r_i^{(k)}}{\beta + N_{k-1} + m_k}. \tag{10}$$

For a correct response at step k, the reward is μ_{k-1}; this also covers $k = 1$ because $r_i^{(1)} = \delta_i^{(1)}$ and $\mu_0 = 1$. Therefore,

$$\sum_{i \in \mathcal{A}_k} r_i^{(k)} = \mu_{k-1} \sum_{i \in \mathcal{A}_k} \delta_i^{(k)} = \mu_{k-1} m_k p_k. \tag{11}$$

Substitution gives

$$\mu_k = \mu_{k-1} \frac{\beta + N_{k-1} + m_k p_k}{\beta + N_{k-1} + m_k}. \tag{12}$$

Because $0 \le p_k \le 1$, the sequence $\{\mu_k\}$ is non-increasing. Moreover, $\beta > 0$ ensures $\mu_k > 0$ even when all observed rewards are zero. For a correct sample i at step $k \le t$, $r_i^{(k)} = \mu_{k-1} \ge \mu_t$. Hence its centered advantage $a_i^{(k)} = r_i^{(k)} - \mu_t$ is non-negative. Since $\sigma_t + \epsilon > 0$, the normalized advantage has the same sign, proving the first property.

For the second property, consider correct samples at steps $k < l \le t$. Because a sample remains active at step l, every preceding active set through step $l-1$ contains at least one failure; thus $p_h < 1$ for $k \le h < l$. Since $\beta > 0$ implies $\mu_{k-1} > 0$, the recurrence above is strictly decreasing over these steps. Consequently,

$$r_i^{(k)} - r_j^{(l)} = \mu_{k-1} - \mu_{l-1} > 0. \tag{13}$$

Both responses are normalized with the same μ_t and σ_t, so $\hat{a}_i^{(k)} > \hat{a}_j^{(l)}$.

For the third property, a correct sample at step t has reward μ_{t-1}. Using the recurrence for μ_t, its centered advantage is

$$a_i^{(t)} = \mu_{t-1} - \mu_t = \frac{\mu_{t-1} m_t}{\beta + N_{t-1} + m_t}(1 - p_t). \tag{14}$$

Therefore, as $p_t \to 1$, we have $a_i^{(t)} \to 0$. Since $\sigma_t + \epsilon \ge \epsilon > 0$, the normalized advantage also satisfies $\hat{a}_i^{(t)} \to 0$. □

B Detailed Experimental Settings

B.1 Example Interaction Trace

The following trace shows one SocraticPO interaction from the Material domain. Ellipses indicate text omitted for brevity while preserving the original dialogue structure.

Task instruction. Given a question and four options, select the right answer. Respond with `<reasoning> ... </reasoning>` and `<answer> ... </answer>`. For the answer, only output the letter corresponding to the correct option.

Question. Based on the given data, determine which material is most favorable for the stability of perovskite materials. The stability can be predicted by element properties such as density and first ionization potential.

Given data.
Y4Sr4Fe2Co6O24: Density 6.07425, FIP 6.95;

Y4Sr4Fe2Ni6O24: Density 6.07425, FIP 6.865625;
Y4Sr4Mn1Co7O24: Density 6.11125, FIP 6.9221875;
Y4Sr4Mn1Ni7O24: Density 7.61475, FIP 8.3265625.

Options.
A: Y4Sr4Mn1Co7O24;
B: Y4Sr4Mn1Ni7O24;
C: Y4Sr4Fe2Co6O24;
D: Y4Sr4Fe2Ni6O24.

Student attempt 1. `<reasoning>` To determine which perovskite material is most favorable for stability, we are given two key properties: density and first ionization potential. ... Therefore, the best choice is C. `</reasoning> <answer>C</answer>`

Teacher feedback 1. Your reasoning contains a critical contradiction in Step 2. You stated that "Material C also has a slightly higher density (6.111) than Material D," but the data table shows Material C has density 6.07425, while Material A is the one with 6.11125. ... *Next Step*: (1) Re-check the density values for each material. (2) Re-evaluate the trend: ...

Student attempt 2. `<reasoning>` I apologize for the earlier error. Let's carefully re-examine the data and correct the reasoning step by step. ... While high density alone is not a guarantee, in combination with high FIP, it suggests a stable, tightly bound structure. Final decision: Material B has the best combination of high ionization potential and high density. Therefore, the most favorable for stability is B. `</reasoning> <answer>B</answer>`

Verifier outcome. Attempt 1 is incorrect and attempt 2 is correct. The trajectory is solved on the second complete student attempt.

B.2 Teacher Prompt Templates

You are an expert teacher. Student's answer is incorrect. Please review the student's latest answer and provide concise, actionable guidance. Prefer hints, corrections, and next-step suggestions over directly revealing the answer.

TEACHING PRINCIPLE:

- Point out the main issue in the student's answer.
- Give the next step or correction they should try.
- Do not solve the full problem for them unless explicitly allowed.
- Keep the guidance short and concrete.

Question: {question}
Student's latest answer: {student_answer}
Reference solution: {reference_solution}
Ground-truth block: {ground_truth_block}
Output Format: Write a short piece of guidance for the student.

B.3 Dataset Statistics

We use the reasoning subsets of SciKnowEval in four scientific domains and additionally evaluate mathematical reasoning on the Math benchmark. Table 3 reports the number of training and evaluation examples in each benchmark.

Table 3: Dataset statistics for the five evaluation benchmarks.

Domain	Train examples	Evaluation examples	Total
Chemistry	1,890	210	2,100
Physics	720	80	800
Biology	450	50	500
Material	841	94	935
Math	7,473	1,319	8,792

Table 4 shows representative examples from the five benchmarks. The four SciKnowEval tasks are multiple-choice questions evaluated by the final option letter, whereas Math uses an open-ended numerical answer.

Table 4: Representative examples from the five evaluation benchmarks.

Benchmark	Question (excerpt)	Answer
Chemistry	Which of the following molecules is the only correct reactant used in the synthesis of the product "Nc1ccccc1S(=O)(=O)N1CC=CC1" ? A: O=[N+]([O-])c1ccccc1S(=O)(=O)N1CC=CC1 B: O=[N+]([O-])c1ccccc1S(=O)(=O)NC1CCCCC1 C: O=[N+]([O-])c1ccccc1S(=O)(=O)Cl D: O=[N+]([O-])c1ccccc1S(=O)(=O)NC1CCC1	B
Physics	What is the rotational period of the Milky Way's galactic halo, given its average radius of 77122 light-years? Please note that the following content involves constants might be used in the calculations :pi = 3.14 ; velocity of light: c = 3e8 m/s ;light year = 9.461e15 m. Additionally, please select the option from the given choices that you believe is closest to the correct answer! A: 6.61e+08 years B: 4.97e+08 years C: 1.7e+08 years D: 4.1e+08 years	B
Biology	With a fitness of 1.0 for the wide-type protein sequence 'MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGVDGEWTYDDATKTFTVTE', a four-sites random mutation has been carried out to produce the mutant 'MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGQWMEWTYDDATKTFTHTE'. What is the new fitness value of this mutant? A: 0.01 B: 0.022 C: 0.0 D: 2.774	B
Material	Given the following crystal structure parameters for the material RbLa9(IrO6)4 (Material ID: mp-560657), calculate the volume of the unit cell (in $Å^3$). Lattice: $a = 7.82$, $b = 7.82$, $c = 17.88$ Å, $\alpha = \beta = \gamma = 90^\circ$. A: 1025.67 B: 1094.31 C: 1200.45 D: 1150.78	B
Math	Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and uses four to bake muffins for her friends. She sells the remainder at the farmers' market for $2 per egg. How many dollars does she make each day?	18

B.4 Implementation Details

Table 5 summarizes the main training and inference settings in a grouped format. All experiments are conducted on a 6-card H100 GPU machine. We allocate 4 GPUs for student-policy training and 2 GPUs for teacher inference during SocraticPO rollouts, so that teacher feedback can be generated online without blocking the student update pipeline as much as possible.

Table 5: Hyperparameters for Reinforce++, GRPO, GSPO, SDPO, and SocraticPO. Entries marked "–" indicate parameters not applicable to that method.

Parameters	Reinforce++	GRPO	GSPO	SDPO	SocraticPO
General					
Student model	Qwen3-4B-Inst	Qwen3-4B-Ins	Qwen3-4B-Ins	Qwen3-4B-Ins	Qwen3-4B-Ins
Teacher model	–	–	–	Qwen3-4B-Ins	Qwen{3-4B-Ins,3.5-27B}, Gemma3-27B
Thinking mode	False	False	False	False	False
Data					
Max. prompt length	8192	8192	8192	8192	8192
Max. response length	8192	8192	8192	8192	8192
Max. teacher feedback length	–	–	–	–	1024
Batching					
Question batch size	32	32	32	32	32
Rollout batch size	–	–	256	–	–
Mini batch size	8	8	8	8	8
Number of rollouts	8	8	8	8	8
Maximum interaction steps K	–	–	–	–	2 or 3
Rollout					
Inference engine	VLLM	VLLM	VLLM	VLLM	VLLM
Student temperature	1.0	1.0	1.0	1.0	1.0
Student top-p	–	–	1.0	–	–
Teacher temperature	–	–	–	1.0	0.2
Validation					
Number of rollouts	16	16	16	16	16
Temperature	0.6	0.6	0.6	0.6	0.6
Top-p	0.95	0.95	0.95	0.95	0.95
RL loss					
Importance-ratio type	–	–	Segment	–	–
Use sequence clip range	–	–	True	–	–
Sequence clip (low/high)	–	–	0.0003 / 0.0004	–	–
Dual-clip loss	–	–	False	–	–
ϵ-high asymmetric clip	–	0.28	–	–	0.28
Rollout IS clip (ρ)	–	2	–	–	2
KL coefficient	0.0	0.0	0.0	–	0.0
Token-level KL / KL loss	–	–	False / False	–	–
Reference model	–	–	False	–	–
Training					
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Learning rate	1×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{-5}	1×10^{-6}
Warmup steps	10	10	10	10	10
Weight decay	0.01	0.01	0.01	0.01	0.01
Gradient clip norm	1.0	1.0	1.0	1.0	1.0